

Docket No.: K-0329

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Myung Sub SIM

New U.S. Patent Application

Filed: October 16, 2001

For: DEVICE AND METHOD FOR DECODING TURBO CODES

TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Assistant Commissioner of Patents  
Washington, D. C. 20231

Sir:

At the time the above application was filed, priority was claimed based on the following application:

Korean Patent Application No. 60746/2000, filed October 16, 2000.

A copy of each priority application listed above is enclosed.

Respectfully submitted,  
FLESHNER & KIM, LLP



Daniel Y.J. Kim  
Registration No. 36,186  
David W. Ward  
Registration No. 45,198

P. O. Box 221200  
Chantilly, Virginia 20153-1200  
703 502-9440

**Date: October 16, 2001**

DYK/DWW:tmd



1C997 U.S. PTO  
09/977252  
10/16/01

# 대한민국 특허청

## KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원번호 : 특허출원 2000년 제 60746 호  
Application Number

출원년월일 : 2000년 10월 16일  
Date of Application

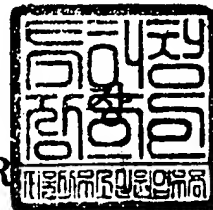
출원인 : 엘지전자 주식회사  
Applicant(s)

**CERTIFIED COPY OF  
PRIORITY DOCUMENT**

2001 년 06 월 13 일

특 허 청

COMMISSIONER



【서류명】	특허출원서		
【권리구분】	특허		
【수신처】	특허청장		
【참조번호】	0007		
【제출일자】	2000.10.16		
【국제특허분류】	H04B		
【발명의 명칭】	터보 디코딩 방법		
【발명의 영문명칭】	Method for decoding Turbo		
【출원인】			
【명칭】	엘지전자 주식회사		
【출원인코드】	1-1998-000275-8		
【대리인】			
【성명】	김용인		
【대리인코드】	9-1998-000022-1		
【포괄위임등록번호】	2000-005155-0		
【대리인】			
【성명】	심창섭		
【대리인코드】	9-1998-000279-9		
【포괄위임등록번호】	2000-005154-2		
【발명자】			
【성명의 국문표기】	심명섭		
【성명의 영문표기】	SIM, Myung Sub		
【주민등록번호】	681008-1056913		
【우편번호】	150-050		
【주소】	서울특별시 영등포구 신길동 347-269		
【국적】	KR		
【심사청구】	청구		
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 김용인 (인) 대리인 심창섭 (인)		
【수수료】			
【기본출원료】	20	면	29,000 원
【가산출원료】	0	면	0 원

1020000060746

2001/6/1

【우선권주장료】	0	건	0	원
【심사청구료】	7	항	333,000	원
【합계】	362,000			원
【첨부서류】	1.	요약서·명세서(도면)_1통		

**【요약서】****【요약】**

본 발명은 차세대 이동통신에 관한 것으로, 특히 슬라이딩 윈도우 방식을 이용한 터보 코드의 디코딩 방법에 관한 것이다. 이와 같은 본 발명에 따른 터보 디코딩 방법은 수신 시퀀스를 최대 사후(Maximum A Posteriori) 알고리즘을 이용하여 디코딩하는데 있어, 일정 길이동안 역방향 프로세싱에 의한 학습(learning)을 수행하고, 이후에 역방향 프로세싱에 의한 제1 결과값을 계산하고, 이와 동시에 순방향 프로세싱에 의한 제2 결과값을 계산하여, 상기 제1 결과값 이전에 저장된 제1 결과값을 이용하여 디코딩 비트 출력을 결정하여 이루어진다. 따라서, 발명은 역방향 프로세서를 하나만 사용하므로 회로의 크기와 연산량(전력소모)을 줄일 수 있고, 일정한 길이 단위로 수행되어지는 프로세싱의 마지막 윈도우를 채움으로써 트렐리스 터미네이션의 특성을 살려 더 나은 디코딩 능력을 갖는다.

**【대표도】**

도 2

**【색인어】**

순방향(알파) 프로세서, 역방향(베타) 프로세서, 학습(learning)

**【명세서】****【발명의 명칭】**

터보 디코딩 방법{Method for decoding Turbo}

**【도면의 간단한 설명】**

도 1은 종래 기술에 따른 MAP 디코딩 타이밍도를 나타낸 도면.

도 2는 본 발명에 따른 MAP 디코딩 타이밍도를 나타낸 도면.

**【발명의 상세한 설명】****【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

- <3> 본 발명은 차세대 이동통신에 관한 것으로, 특히 슬라이딩 윈도우 방식을 이용한 터보 코드의 디코딩 방법에 관한 것이다.
- <4> 알려진 바와 같이 터보 코드는 두 개의 순환 계통적 컨벌루션 부호화기(RSC ; Recursive Systematic Convolutional Encoder)가 내부 인터리버(Interleaver)를 통하여 병렬적으로 연결되어 생성된 부호로써, 차세대 이동 통신 규격에서 고전송율의 데이터를 전송하기 위한 코딩 방식으로 이용되고 있다.
- <5> 이러한 터보 코드는 생성된 정보 비트열에 대하여 블록 단위로 처리하고 있는데, 특히 큰 정보 비트열을 인코딩하는 경우 컨벌루션 부호에 대하여 매우 우수한 코딩이득을 가지고 있으며, 수신단에서 간단한 성분코드에 대한 디코딩을 반복적으로 행함으로써 매우 우수한 오류 정정 능력을 가지는 것으로 알려져 있다.

- <6> 최근에는 이동통신 환경 하에서 고속 데이터 전송을 지원할 수 있는 비교적 간단한 터보 디코딩 기법이 제안되었는데, 이 구조에서는 입력되는 수신 코드워드가 두 개의 컨벌루션 디코더를 교대로 통과하도록 하여 그 구조의 복잡도를 상당히 줄였다. 그러나, 반복적으로(iteratively) 컨벌루션 디코더를 통과시키기 위해서는 컨벌루션 디코더의 출력이 '0' 또는 '1'로 경판정(hard decision)된 값이 아니라 '0'이거나 '1'일 확률의 비에 해당하는 소프트한 값이 요구된다. 이를 위해 정보 비트의 사후(A Posteriori) 확률값을 계산하여, 그 확률값이 최대가 되도록 복호하는 최대 사후(Maximum A Posteriori; 이하 MAP이라 약칭함) 디코딩 기법이 제안되었다.
- <7> 일반적으로 터보 부호의 정보원은 불연속 시간과 한정된 상태를 갖는 '마코프 프로세스(Markov process)'이다. 따라서, 정보원은 이진 격자도로 설명될 수 있다.
- <8> 이진 격자에서  $S_k$ 는 시간  $k$ 에서의 부호화기의 메모리 상태를 나타내고 ' $x_k = x_{k,1,k}, x_{k,n}$ '는 부호율이  $1/n$ 인 부호화기의 출력을 나타낸다. ( $x_k = \{0,1\}$ ). 여기서 정보원의 상태 ' $S_k = m$ '는 모두  $M$ 가지의 상태들을 갖는다. ( $m = 0, 1, 2, \dots, M-1$ ) 시간이  $k-1$ 에서  $k$ 로 천이할 때 터보 부호화기의 입력 비트  $d_k$ 는 부호화기의 상태  $S_{k-1}$ 를  $S_k$ 로 변환시킨다. 정보의 상태 시퀀스  $S = (S_0, \dots, S_T)$ 는 시간  $k=0$ 에서 시작해서 시간  $k=T$ 에서 종료된다. 이때 부호화기의 초기 상태인  $S_0$ 는 제로이다.
- <9> 상기 터보 부호화기의 출력 시퀀스  $x$ 는 BPSK 또는 QPSK로 변조되고, 이산 메모리 채널에서 페이딩을 겪는다. 그러므로, 수신단에서 수신된 시퀀스는 ' $y = (y_{1,k}, y_{k,k}, y_{L,k})$ '가 된다. 여기서  $y_k = (y_{k,1,k}, y_{k,n,k})$ 이다.
- <10> 따라서, MAP 알고리즘은 상기 수신된 시퀀스를 이용해서 정보의 상태 천이 사후확

를 추정하기 위한 알고리즘이다. 또한 정보 비트의 사후확률  $P(d_k=1 | y)$ 와  $P(d_k=0 | y)$ 를 계산해야 한다. 그러면, 최종적으로 원하는 LLR(log likelihood ratio) 형태의 복호화의 출력을 구할 수 있다.

<11> 【수학식 1】

$$L(d_k) = \log \frac{P(d_k=1 | y)}{P(d_k=0 | y)}$$

$$= \log \frac{\sum_{(m', m) d_k=1} P(S_{k-1}=m', S_k=m, y)}{\sum_{(m', m) d_k=0} P(S_{k-1}=m', S_k=m, y)}$$

<12> 이때, 정보 비트의 상태 천이 사후확률  $P(S_{k-1}=m', S_k=m, y)$ 은 다음 수학식 2와 같이 구해진다.

<13> 【수학식 2】

$$P(S_{k-1}=m', S_k=m, y) = P(S_{k-1}=m', y_{j < k}) P(y_{j > k} | S_k=m)$$

<14> 
$$P(S_k=m', y_k | S_{k-1}=m')$$

<15> 상기 수학식 2에서  $y_{j < k}$ 는 처음 시간부터 k-1까지의 수신 시퀀스  $y_j$ 를 나타내며,  $y_{j > k}$ 는 시간 k+1부터 마지막 수신 시퀀스까지를 나타낸다.

<16> 상기 수학식 2에서  $P(S_{k-1}=m', y_{j < k})$ 는 ' $\alpha(S_{k-1})$ '로 정의되고, 이로부터 ' $\alpha(S_k)$ '가 정의되고, 상기  $P(y_{j > k} | S_k=m)$ 는 ' $\beta(S_k)$ '로 정의된다.

<17> 최적의 사후 확률을 얻기 위해서는 상기 ' $\beta(S_k)$ '을 구하기에 앞서 일정 길이동안의 기간이 필요하다. 이를 '학습' 기간이라고 하고, 이 학습 기간 이후에 계산되어진 ' $\beta(S_k)$ '가 디코더 출력 비트 결정을 위해 이용된다.

<18> 이하 상기  $\alpha(S_k)$ 는 알파값, 상기  $\beta(S_k)$ 는 베타값으로 지칭된다.



- <19> 도 1은 종래 기술에 따른 MAP 디코딩 타이밍도를 나타낸 도면이다.
- <20> 도 1에서 X축은 시간의 흐름을 나타내며, 각 프로세서가 시간의 흐름에 따라 어느 심볼을 처리하는가를 나타낸 것이다. 순방향 프로세서의 심볼의 번호는 증가하며 역방향 프로세서의 심볼 번호는 감소한다. 그리고, 빗금친 구간에서 역방향 프로세서는 '학습' 중임을 나타낸다. 그리고, 곡선의 화살표는 비트 결정에 필요한 알파, 베타값들의 의존도를 보이고 있다.
- <21> 도 1을 참고하면, 상기 역방향 프로세서 두 개를 사용하여 제1 역방향 프로세서가 '학습(learning)'을 하는 동안 제2 역방향 프로세서가 디코더의 비트 결정에 필요한 베타값을 계산한다. 즉, 하나의 MAP 디코딩이 시작되면 제1 역방향 프로세서가 2L부터 1L까지 '학습' 과정을 수행한다. 이 동안 제2 역방향 프로세서는 정지 상태로 있다. 여기서 L은 수신된 시퀀스의 노드 길이를 나타낸다.
- <22> 이후에 제1 역방향 프로세서는 1L부터 0까지의 베타값을 계산하며, 이미 계산되어 저장된 0~1L까지의 알파값을 함께 사용하여 1L부터 0까지의 디코더의 비트를 결정한다. 이 시간동안 제2 역방향 프로세서는 3L부터 2L까지의 심볼을 이용하여 '학습'을 수행한다.
- <23> 다음 구간에서는 제2 역방향 프로세서가 2L부터 1L까지의 베타값을 계산하여 2L부터 1L까지의 디코더의 비트를 결정한다. 이 동안 제1 역방향 프로세서는 4L부터 3L까지의 심볼을 이용하여 '학습'을 수행한다.
- <24> 디코더 출력 블록에서 볼 수 있듯이, 비트 결정은 1L부터 0까지, 2L부터 1L까지 순서로 이루어지므로 L만큼씩 출력을 저장했다가 끝에서부터 읽어내는 후입선출(Last In

First Out : LIFO) 과정을 거쳐 바른 순서의 결과를 얻게 된다.

<25> [참고문헌]

<26> [1](A.J Viterbi, 'An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes' IEEE Journal on Selected Areas in Communications, vol.16, no.2, Feb. 1998.)

<27> 이와 같은 종래 기술에서는 거의 모든 심볼에 대해서 역방향 프로세싱을 두 번씩 수행한다. 결과적으로 매번의 MAP 디코딩마다 두 번의 역방향 프로세싱이 필요하게 되므로 연산량이 증가하게 되고, 이는 전력소모를 증가시켜 배터리로 동작하는 무선 이동장비의 사용시간을 줄이는 문제점이 있다.

<28> 또한, 연산량을 감소시키기 위하여 하나의 역방향 프로세서만을 사용하게 되는 경우에는 디코딩 타임이 두 배로 증가하는 문제점이 있다.

<29> 또한, L길이동안 학습 과정과, 순방향 프로세싱과, 역방향 프로세싱을 수행하게 되면, 터보 코드의 트렐리스 중단( $S_T=0$ ) 상태에서 완료되는 특성을 충분히 살리지 못해서, 터보 코드의 코딩 이득이 저하되는 문제점도 발생된다.

<30> 또한, 상기 결과값을 저장하기 위한 메모리의 크기가 'depth 60\*width 56(3GPP WCDMA 터보 부호기 가정)'으로 작긴 하지만 요즘 일반적으로 사용하는 메모리의 깊이(depth)는 이보다 훨씬 커서 많은 메모리를 낭비하게 된다.

#### 【발명이 이루고자 하는 기술적 과제】

<31> 따라서, 본 발명의 목적은 이상에서 언급한 종래 기술의 문제점을 감안하여 안출한 것으로서, 하나의 역방향 프로세서를 이용하여, 적은 연산량과 작은 메모리를 요구하는

터보 디코딩 방법을 제공하기 위한 것이다.

<32> 또한, 본 발명의 다른 목적은 하나의 역방향 프로세서를 이용하여 트렐리스 종단을 최대한 활용하는 터보 디코딩 방법을 제공하기 위한 것이다.

<33> 이상과 같은 목적을 달성하기 위한 본 발명의 방법상 특징에 따르면, 수신 시퀀스를 최대 사후(Maximum A Posteriori) 알고리즘을 이용하여 디코딩하는데 있어, 일정 길이 동안 역방향 프로세싱에 의한 학습(learning)을 수행하고, 이후에 역방향 프로세싱에 의한 제1 결과값을 계산하고, 이와 동시에 순방향 프로세싱에 의한 제2 결과값을 계산하여, 상기 제1 결과값 이전에 저장된 제1 결과값을 이용하여 디코딩 비트 출력을 결정한다.

<34> 바람직하게, 상기 역방향 또는 순방향 프로세싱 시간이  $W$ 이고, 상기 수신 시퀀스 길이를  $W$ 로 나눈 나머지가  $W_0$ 이고, 정수  $N$ 에 대하여, 수신 시퀀스의 시간 ' $W_0 + NW + L$ '부터 ' $W_0 + NW$ '의 심볼로 역방향 프로세싱에 의해 학습(learning)을 수행하고, 이후에 ' $W_0 + NW$ '부터 ' $W_0 + (N-1)W$ '까지의 심볼로 역방향 프로세싱에 의한 제1 결과값을 저장하고, 이와 동시에  $0(N=1)$  또는 ' $W_0 + (N-2)W$ '( $N$ 은 2이상의 정수)부터 ' $W_0 + (N-1)W$ '까지의 심볼로 순방향 프로세싱에 의한 제2 결과값을 계산하여, ' $W_0 + (N-1)W$ '부터 ' $W_0 + (N-2)W$ '까지 계산되어 저장된 제1 결과값과 함께 디코딩 비트 결정을 수행한다.

<35> 한편, 상기  $N$ 이 0인 경우에, 상기 수신 시퀀스의 시간 ' $W_0 + L$ '부터 ' $W_0$ '의 심볼로 역방향 프로세싱에 의해 학습(learning)을 수행하고, 이후에 ' $W_0$ '부터 0까지의 심볼로 역방향 프로세싱에 의한 제1 결과값을 저장한다.

<36> 상기 제1 결과값은 듀얼 포트 램(DPRAM)의 어느 한 포트를 통하여 저장되고, 또 다

른 포트를 통하여 읽혀지는데, 상기 듀얼 포트 램(DPRAM)의 포트들을 통해 저장 또는 읽혀지는 주소는 상기  $W_0$  또는  $W$  길이마다 증가 또는 감소된다.

<37> 또한, 상기 듀얼 포트 램(DPRAM)의 포트들을 통해 저장 또는 읽혀지는 주소는 상기  $W_0$  또는  $W$  길이마다 상호 배타적으로 증가 또는 감소된다.

<38> 또한, 상기 디코딩 비트의 결정 출력이 순서대로 이루어지므로 후입선출(Last In First Out)이 제거되는 것을 특징으로 한다.

#### 【발명의 구성 및 작용】

<39> 이하 본 발명의 바람직한 일 실시 예에 따른 구성 및 작용을 첨부된 도면을 참조하여 설명한다.

<40> 앞에서 설명한 바와 같이 MAP 알고리즘은 수신된 시퀀스( $y_k$ )를 이용해서 정보 비트의 사후확률(A Posteriori) [ $P(d_k=1 | y)$ 와  $P(d_k=0 | y)$ ]을 추정하고, 이 정보의 사후확률로부터 최종적으로 LLR(Log Likelihood Ratio) 형태의 복호화기의 출력을 구하기 위한 알고리즘이다. 상기 LLR 형태의 복호화기의 출력은 수학적 식 1을 통하여 설명되어져 있다.

<41> 이때, 정보 비트의 사후 확률을 구하기 위해서는 ' $P(d_k=1 | y)$ '와 ' $P(d_k=0 | y)$ ' 각각에 대해 상태 천이 사후확률을 구해야 하는데, 이 각각의 상태 천이 사후확률은 세 가지의 곱셈 항목에 의하여 구할 수 있다. 상기 상태 천이 사후확률에 대한 세 가지의 곱셈 항목은 수학적 식 2를 통하여 설명되어져 있다.

<42> 즉, 수학적 식 2를 참고로 하면 제1 항목( $\alpha(S_{k-1})$ )은 수신 시퀀스가 시간 0부터  $k-1$ 에 해당하는 확률이고, 상태  $S_{k-1}$ 이  $m'$ 을 갖는 조인트 확률 함수로 표현되고, 다음 수학적 식 3과 같이 나타낸다.

## &lt;43&gt; 【수학식 3】

$$\alpha(S_{k-1}) = P(S_{k-1} = m' | y_{j < k})$$

<44> 상기 수학식 3으로부터  $\alpha(S_k)$ 는 시간 1부터 k를 갖는 수신 시퀀스( $y_{j < k+1}$ )에서, 상태  $S_{k-1}$ 이  $m'$ 이고, 상태  $S_k$ 가  $m$ 인 상태천이에 대한 조인트(Joint) 확률 밀도 함수로 표현이 되고, 다음 수학식 4와 같이 나타낸다.

## &lt;45&gt; 【수학식 4】

$$\alpha(S_k) = \sum_{m'=0}^{M-1} P(S_{k-1} = m', S_k = m | y_{j < k+1})$$

<46> 제2 항목( $\gamma(S_{k-1}, S_k)$ )은 상태  $S_{k-1}$ 가 상태  $S_k$ 로 천이할 때 관계하는 가지 메트릭(branch metric)으로써, 상태  $S_{k-1}$ 이  $m'$ 인 조건하에서 다음 상태  $S_k$ 가  $m$ 이고, 이때 수신된 시퀀스는  $y_k$ 를 갖는 조건 확률 함수로 표현이 되고, 다음 수학식 5와 같이 나타낸다.

## &lt;47&gt; 【수학식 5】

$$\gamma(S_{k-1}, S_k) = P(S_k = m | y_k | S_{k-1} = m')$$

<48> 제3 텀( $\beta(S_k)$ )은 상태  $S_k$ 가  $m$ 인 조건하에서 수신 시퀀스 넘버가  $k+1$  이상이 되는 수신 시퀀스( $y_{j > k}$ )가 될 조건 확률 함수로 표현이 되고, 다음 수학식 6과 같이 나타낸다

## &lt;49&gt; 【수학식 6】

$$\beta(S_k) = P(y_{j > k} | S_k = m)$$

<50> 상기  $d_k$ 는 터보 부호화되기 전의 정보 비트열을 나타내고,  $S_k$ 는 시간 k에서의 부호화의 메모리 상태( $m = \{0, 1, \dots, M-1\}$ )를 나타내는 것으로, 모두 M가지의 상태를 가지며, 시간이 k-1에서 k로 천이할 때 상기 입력비트  $d_k$ 는 부호화기의 상태  $S_{k-1}$ 를  $S_k$ 로 변환된

다.

<51> 즉, MAP 디코딩에서 상기 ' $\alpha(S_k)$ '는 수학식 4와 같이 순방향 재귀(forward recursion) 방법에 의하여 구할 수 있으며, 계산된 알파값이 디코더의 출력 비트를 결정하는데 바로 이용된다. 상기 ' $\alpha(S_k)$ '는 하나의 순방향(알파) 프로세서에 의하여 수행된다.

<52> 또한, 상기 ' $\beta(S_k)$ '는 수학식 6과 같이 역방향 재귀(backward recursion) 방법에 의하여 구할 수 있으며, 이 ' $\beta(S_k)$ '가 최적의 사후 확률을 구하는데 이용되기 위해서는 일정 기간이 필요하다. 이 일정 기간을 '학습(learning)' 기간이라고 하고, 이 학습 기간 이후에 한 개의 역방향(베타) 프로세서에 의하여 계산되어진 ' $\beta(S_k)$ '가 디코더의 출력 비트를 결정하는데 이용된다.

<53> 이하 상기  $\alpha(S_k)$ 는 알파값, 상기  $\beta(S_k)$ 는 베타값으로 지칭된다.

<54> 도 2는 본 발명에 따른 MAP 디코딩 타이밍도를 나타낸 도면이다.

<55> 도 2를 참조하면, 역방향 프로세서는 임의의  $W_0+L$ 부터  $W_0$ 까지의 심볼을 가지고 '학습'을 시작한다. 상기 '학습'에 대한 기간은 전체 MAP 디코딩에서 'L' 길이동안 이루어진다. 상기 '학습'에는 종래 기술과 같이 수신 시퀀스의 L비트에 해당하는 심볼을 사용한다.

<56> 다음  $W_0$ 에서부터 0까지의 심볼을 가지고 베타값을 계산하여 메모리에 저장한다.

<57> 다음으로  $W_0+L+W$ 부터  $W_0+W$ 까지의 심볼을 가지고 '학습' 과정을 시작하고,  $W_0+W$ 부터  $W_0$ 까지의 심볼로 베타값을 계산하여 메모리에 저장한다. 이와 동시에 순방향 프로세서는 0에서  $W_0$ 까지의 심볼을 가지고 알파값을 계산하여, 상기 역방향 프로세서에 의해 저장

된  $W_0$ 에서 0까지의 베타값을 같이 이용하여 디코더 출력의 비트를 결정한다. 상기 베타값 또는 알파값의 계산은  $W$  길이동안 이루어진다.

<58> 계속해서  $W_0+2W+L$ 부터  $W_0+2W$ 까지의 심볼을 가지고 '학습' 과정을 시작하고,  $W_0+2W$ 부터  $W_0+W$ 까지의 심볼로 베타값을 계산하여 메모리에 저장한다. 이와 동시에 순방향 프로세서는 상기  $W_0$ 부터  $W_0+W$ 까지의 심볼을 가지고 알파값을 계산하여, 상기 역방향 프로세서에 의해 저장된  $W_0+W$ 부터  $W_0$ 까지의 베타값을 같이 이용하여 디코더 출력의 비트를 결정한다.

<59> 이와 같이 '학습'과, 역방향 프로세싱, 순방향 프로세싱, 디코딩을 반복해서 하나의 코드 블록에 대한 MAP 디코딩을 완료한다.

<60> 한편, 도 2의 아래 두 줄에는 역방향 프로세싱의 결과인 베타값을 저장하는 듀얼 포트 램(RAM)(DPRAM)의 어드레스를 나타내었다. 포트 A로는 베타값을 저장하고, 포트 B로는 베타값을 읽어내는 것을 가정하였다.

<61> 단, 상기 '학습'기간 중에 계산되어진 베타값은 저장하지 않는다.

<62> 상기  $W_0$ 부터 0까지 계산된 베타값은 포트 A의  $W_0$ 부터 0까지 감소하는 어드레스의 순서대로 저장되고, 이후에 상기 베타값은 포트 B를 통하여 0부터  $W_0$ 까지 증가하는 어드레스 순서대로 읽어져 상기 0부터  $W_0$ 까지 계산된 알파값과 함께 디코더의 출력 비트를 결정하는데 이용된다.

<63> 또한, 상기 ' $W_0+W$ '부터 ' $W_0$ '까지 계산된 베타값은 포트 A의 0부터  $W$ 까지 증가하는 어드레스의 순서대로 저장되고, 이후에 상기 베타값은 포트 B를 통하여  $W$ 부터 0까지 감소하는 어드레스 순서대로 읽어져 상기 ' $W_0$ '부터 ' $W_0+W$ '까지 계산된 알파값과 함께 디코더의

출력 비트를 결정하는데 이용된다.

- <64> 즉, 상기 포트 A와 포트 B는 상기 일정 길이동안 계산되어진 베타값을 증가 또는 감소하는 어드레스의 순서대로 저장하거나 읽어서 출력하는데, 상기 포트 A에 저장된 베타값이 증가되는 어드레스 순서대로 저장된 경우에는 포트 B를 통하여 감소하는 어드레스 순서대로 읽어지고, 감소되는 어드레스 순서대로 저장된 경우에는 포트 B를 통하여 증가하는 어드레스 순서대로 읽어져서 출력된다.
- <65> 즉, 상기 포트 A와 포트 B는 상호 배타적으로 증가 또는 감소하는 어드레스 순서에 따라 동작한다.
- <66> 만약, 쓰고 있는 어드레스가 항상 같은 방향으로 진행된다면 즉, 포트 A의 어드레스는  $W$  또는  $W_0$ 에서 0으로 감소하고 포트 B의 어드레스는 0에서  $W$  또는  $W_0$ 로 증가하기를 반복한다면, 저장된 베타값을 채 읽어내지 않은 상태에서 새로운 값이 갱신되어 버린다. 이를 방지하려면 베타 저장 메모리를 두 배로 늘리거나, 본 발명에서 제안된 방법을 사용해야 하는 것이다.
- <67> 본 발명에서는 하나의 역방향 프로세싱이  $W_0 + NW + L$  ( $N=0, 1, 2, \dots$ )에서 시작되므로 이 시작점은 슬라이딩 윈도우마다  $W$ 만큼씩 증가한다. 그리고, 역방향 프로세싱의 끝점은  $0(N=0)$  또는  $W_0 + (N-1)W$  ( $N=1, 2, \dots$ )으로써 역시  $W$ 만큼씩 증가시키면 된다. 모든 '학습'에는 종래 기술과 같이 수신 시퀀스의  $L$ 비트에 해당하는 심볼을 사용한다.
- <68> 상기  $W_0$ 는 (수신 시퀀스 길이)% $W$  (%는 모듈로 연산)로 결정한다. 단, 모듈로 결과가 0이면  $W$ 를 사용한다.
- <69> 예를 들어, 수신 시퀀스 길이가 3840비트이고,  $W$ 가 256이라면  $W_0$ 는  $W$ 와 같이 256으



로 결정된다. 수신 시퀀스 길이가 3841비트라면  $W_0$ 는 1이 된다. 이렇게  $W_0$ 를 결정하면 마지막 역방향 프로세싱 단위가 항상  $W$ 가 된다. 이렇게 함으로써 3GPP WCDMA와 같이 트렐리스 터미네이션(3GPP TS25.212 V2.2.1 Turbo Coding section, Oct.1999)을 사용하는 터보 코드의 성질을 최대한 살릴 수 있다.

<70> 이것은 테일 비트를 사용한 컨벌루션 코드를 비터비 디코더로 디코딩할 때 코드 블록의 맨 마지막 역추적 깊이(trace-back depth) 만큼은 스테이트 0에서 시작해서 한꺼번에 디코딩하는 것과 같은 개념이다.

<71> 예를 들어,  $W$ 를 256으로 결정했을 때 3841비트의 코드 블록은 1비트가 남게 되고, 본 발명에서처럼 맨 앞 윈도우에서 1비트를 처리하거나, 아니면 맨 뒤의 윈도우에서 1비트를 처리하는 방법을 생각할 수 있다. 그런데, 각 코드 블록의 끝에서는 터미네이션 특성을 이용할 수 있으므로 이를 이용하여 1비트를 디코딩하는 것보다는 256 비트를 디코딩하는 것이 보다 나은 디코딩 성능을 얻을 수 있는 방법이다.

<72> 한편, MAP 디코딩에서 필요한 메모리의 크기를 비교하면 종래 기술에서는 60depth를 필요로 하는 반면, 본 발명의 디코더는 256depth를 사용하므로 많이 늘어난 것처럼 보이지만, 본 발명의 구현에 사용한 'Xilinx Virtex' 칩의 내부 블록 램(RAM)의 최소 깊이(depth)가 256이므로 실제 회로를 구현하는 관점에서는 차이가 없을 수도 있다.

#### 【발명의 효과】

<73> 이상의 설명에서와 같이 본 발명은 역방향 프로세서를 하나만 사용하므로 회로의 크기와 연산량(전력소모)을 줄일 수 있다.

<74> 마지막 윈도우를 채움으로써 트렐리스 터미네이션의 특성을 살려 더 나은 디코딩

능력을 갖는다.

- <75> 디코더 결과를 순서대로 얻을 수 있으므로 LIFO에 필요한 메모리 및 회로를 제거할 수 있고, 전력 소모면에서 개선효과가 있다.
- <76> 베타값을 저장하는 듀얼 포트 램(DPRAM)의 읽고 쓰는 어드레스를 조절하여 이 메모리의 크기를 반으로 줄일 수 있다.
- <77> 이상 설명한 내용을 통해 당업자라면 본 발명의 기술 사상을 일탈하지 아니하는 범위에서 다양한 변경 및 수정이 가능함을 알 수 있을 것이다.
- <78> 따라서, 본 발명의 기술적 범위는 실시예에 기재된 내용으로 한정하는 것이 아니라 특허 청구 범위에 의해서 정해져야 한다.

## 【특허청구범위】

## 【청구항 1】

- 수신 시퀀스를 최대 사후(Maximum A Posteriori) 알고리즘을 이용하여 디코딩하는데 있어,

일정 길이동안 역방향 프로세싱에 의한 학습(learning)을 수행하고, 이후에 역방향 프로세싱에 의한 제1 결과값을 계산하고, 이와 동시에 순방향 프로세싱에 의한 제2 결과값을 계산하여, 상기 제1 결과값 이전에 저장된 제1 결과값을 이용하여 디코딩 비트 출력을 결정하는 것을 특징으로 하는 터보 디코딩 방법.

## 【청구항 2】

제 1항에 있어서, 상기 역방향 또는 순방향 프로세싱 시간이  $W$ 이고, 상기 수신 시퀀스 길이를  $W$ 로 나눈 나머지가  $W_0$ 이고, 정수  $N$ 에 대하여,

수신 시퀀스의 시간 ' $W_0 + NW + L$ '부터 ' $W_0 + NW$ '의 심볼로 역방향 프로세싱에 의해 학습(learning)을 수행하고, 이후에 ' $W_0 + NW$ '부터 ' $W_0 + (N-1)W$ '까지의 심볼로 역방향 프로세싱에 의한 제1 결과값을 저장하고, 이와 동시에 0( $N=1$ ) 또는 ' $W_0 + (N-2)W$ '( $N$ 은 2이상의 정수)부터 ' $W_0 + (N-1)W$ '까지의 심볼로 순방향 프로세싱에 의한 제2 결과값을 계산하여, ' $W_0 + (N-1)W$ '부터 ' $W_0 + (N-2)W$ '까지 계산되어 저장된 제1 결과값과 함께 디코딩 비트 결정을 수행하는 것을 특징으로 하는 터보 디코딩 방법.

## 【청구항 3】

제 2항에 있어서, 상기  $N$ 이 0인 경우에, 상기 수신 시퀀스의 시간 ' $W_0 + L$ '부터  $W_0$ 의 심볼로 역방향 프로세싱에 의해 학습(learning)을 수행하고, 이후에  $W_0$ 부터 0까지의 심

블로 역방향 프로세싱에 의한 제1 결과값을 저장하는 것을 특징으로 하는 터보 디코딩 방법.

【청구항 4】

제 1항에 있어서, 상기 제1 결과값은 듀얼 포트 램(DPRAM)의 어느 한 포트를 통하여 저장되고, 또 다른 포트를 통하여 읽혀지는 것을 특징으로 하는 터보 디코딩 방법.

【청구항 5】

제 4항에 있어서, 상기 듀얼 포트 램(DPRAM)의 포트들을 통해 저장 또는 읽혀지는 주소는 상기  $W_0$  또는  $W$  길이마다 증가 또는 감소되는 것을 특징으로 하는 터보 디코딩 방법.

【청구항 6】

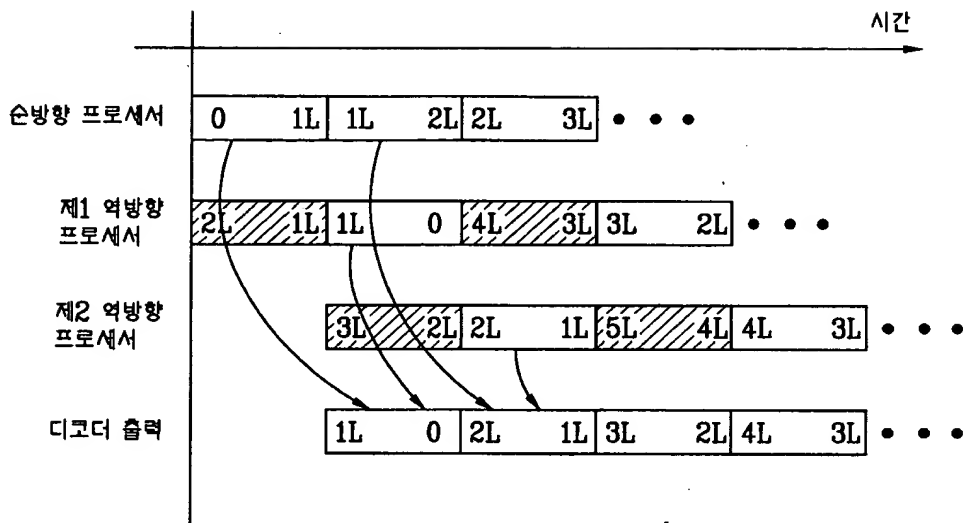
제 4항에 있어서, 상기 듀얼 포트 램(DPRAM)의 포트들을 통해 저장 또는 읽혀지는 주소는 상기  $W_0$  또는  $W$  길이마다 상호 배타적으로 증가 또는 감소되는 것을 특징으로 하는 터보 디코딩 방법.

【청구항 7】

제 1항에 있어서, 상기 디코딩 비트의 결정 출력이 순서대로 이루어지므로 후입선출(Last In First Out)이 제거되는 것을 특징으로 하는 터보 디코딩 방법.

## 【도면】

【도 1】



【도 2】

